Dear Applicant,

First of all, thank you for the interesting conversation in our last talk. We have put together a few things that we think represent the basics of the data-processing/algorithm development at OroraTech well.
For the git part, you may still need to install an appropriate client. Some IDEs (VS Code, pycharm, etc) already have one built in. The satellite data can be downloaded directly from the NASA portal, but for simplicity we also provided the relevant files in Google Drive, so you don't need to register in the NASA portal.

If you have any questions, please feel free to contact us, otherwise we will be happy if you update us as soon as you have found the time to work on the task. You should not have to invest more than one (working) day - although you are of course free to continue working on the project independently of your application, in the spirit of the open source idea.

Assignment Topic: Rudimentary fire detection algorithm for the VIIRS sensor onboard Suomi-NPP satellite

Description:
You will implement a service in **python** that detects thermal anomalies (also “hot pixels” or “hotspots”) in VIIRS images (isn’t that cool!)

Input data:
- Images from the VIIRS instrument onboard the Suomi-NPP satellite from California on the 2020-08-20 and 2020-08-21

All data from this satellite is freely available from NASA earthdata (signup required):
https://search.earthdata.nasa.gov/search
- A Google Drive folder with four sample images that should be enough to get you started:
https://drive.google.com/drive/folders/10A4opHjdt99LrdI_IhKglB3d1G2jWa0X?usp=sharing
- Each satellite overpass consists of two NetCDF files:
    1. File starting with VNP02 contains observation data (radiance recorded by the sensor)
    2. File starting with VNP03 contains geolocation data (the geographical coordinates of each pixel in the observation data file)
- To explore the data you can use the Panoply NetCDF viewer: https://www.giss.nasa.gov/tools/panoply/

Required Output:
- Provide the fire detections as a response to a REST API call (preferrably based on the popular Python library FastAPI). The user uploads a netcdf file (the input data) and receives the fire detections as features in a GeoJSON format: https://geojson.org/

Assumption:
Potential fire pixels (hotspots) are pixels where the “Earth view radiance” in the spectral region between 3.973 μm and 4.128 μm, which is recorded in the **M13** band of the VIIRS sensor, is significantly higher than the nonfire background.

Suggested Workflow:
- Create a Github account and a repository to keep track of your source code
- Write a service to read observation and geolocation data from a single overpass by reading and extracting the relevant information from VNP02 and VNP03 files given to you by a user using the REST interface.
- Derive a threshold for detecting potential fire pixels
- Apply the threshold filter to the MWIR (**M13**) band
- Extract the geolocation of the pixels detected after applying the threshold
- Return the coordinates (latitude,longitude) of the detected pixels (hotspots) to the user as a response to their request

Some more advanced/optional/open-end steps:
- After the detection has been run on the whole image, you can try to associate adjacent "hot pixels" into groups (clusters). One possible thing to do with each group is to save its convex (more difficult: concave) hull in the form of a polygon to a CSV or GeoJSON file.

Additional information:
- General git guide: https://rogerdudler.github.io/git-guide/ (you can ignore the branching, update & merge and tagging chapters)
- More Github specific git steps: https://opensource.com/article/18/1/step-step-guide-git
- VIIRS Fire detection algorithm (Just for reference if you are curious): https://www.researchgate.net/publication/260010654_The_New_VIIRS_375_m_active_fire_detection_data_product_Algorithm_description_and_initial_assessment

Best regards,
OroraTech team